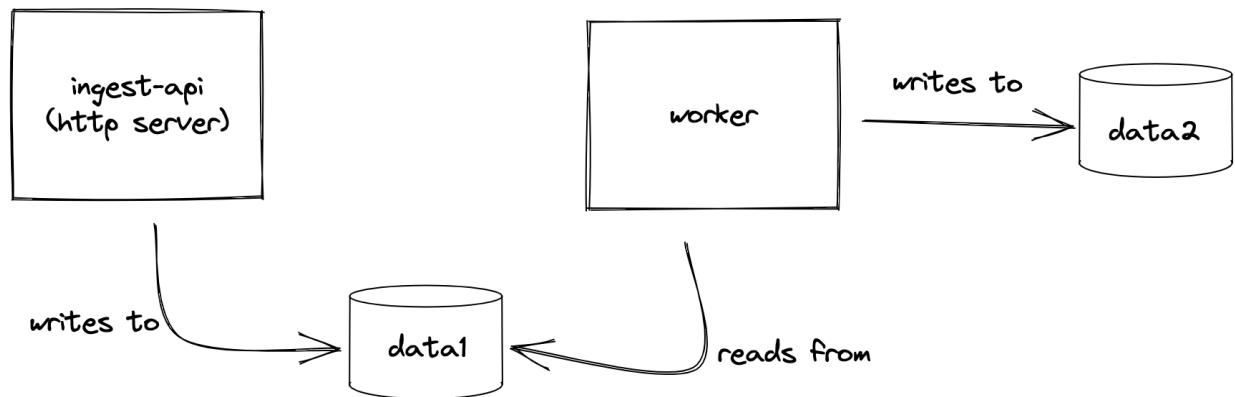


# Take-Home Assignment

At Judo we're building out our analytics offering. The goal is to give users a comprehensive dashboard with an overview of the performance of their experiences. Metrics like average session duration, the number of screens viewed, and the number of users will be displayed.

For your take-home assignment, you're going to build out a mini analytics engine. You're welcome to use any programming language you're comfortable with. You'll implement the following architecture.



The ingest-api will expose a `POST /event` endpoint that receives a minimal JSON body of shape

```
{ "experienceID": <number> }
```

You are required to parse `experienceID` and append that id to a `data1` file. The endpoint may receive the same `experienceID` multiple times, append that id to the `data1` file each time.

You will also build a separate worker which is responsible for periodically parsing the `data1` file by counting how many times each `experienceID` appears and storing the **running count** in a `data2` file. Feel free to store the result in the format of your choosing (e.g, JSON, Python Pickle, etc). The worker should inspect the `MINI_ANALYTICS_INTERVAL` environment variable to determine how often to read from `data1`. `MINI_ANALYTICS_INTERVAL` is given in seconds and if the environment variable is omitted then you should read from `data1` every 5 seconds. Finally, your worker should not read the whole `data1` file each time. Instead, you'll maintain an offset inside the worker and read from that offset.

Please include a `README` describing how to run your solution.

Example:

Suppose we just started our ingest-api server and our worker. `data1` and `data2` are currently both empty files. `MINI_ANALYTICS_INTERVAL` was not set, so our worker will read from `data1` every 5 seconds. Now suppose the following requests come in:

```
POST /ingest-api {"experienceID": 1} POST /ingest-api {"experienceID": 100}
POST /ingest-api {"experienceID": 1}
```

`data1` should then look like:

```
1 100 1
```

Suppose 5 seconds pass and our worker executes. If you're using JSON for your `data2` format, then that file should look like:

```
{ 1: 2, 100: 1 }
```

After the worker completes, another request comes in:

```
POST /ingest-api {"experienceID": 1}
```

`data1` will contain:

```
1 100 1 1
```

After the next execution of the worker, `data2` contain

```
{ 1: 3, 100: 1 }
```

## Submission

You can submit your solution by creating a zip file/tarball and emailing it to [steph@judo.app](mailto:steph@judo.app) or hosting your solution on GitHub and sharing the link.